REMARKS

Claims 1 - 45 are now pending in this application. The Office Action mailed on August 2, 2002 ("Office Action"), rejected Claims 1 - 45 under 35 U.S.C. §103(a) as being unpatentable over European Pat. App. No. 2,296,799 to Platt ("Platt et al.").

Claim Rejections Under 35 U.S.C. §103(a)

The Office Action has rejected claims 1-45 as being unpatentable over Platt and has incorporated the arguments of an Office Action response mailed March 15, 2002. In this rejection, the Office Action states that the Applicant's arguments in the Office Action response mailed on July 15, 2002 were found to be unpersuasive.

Specifically, the first argument, pertaining to storing data from a single key ordered list of objects in contiguous segments as in element (a) of applicant's original claim 1, was rejected. The Office Action reasoned that Platt teaches a single table for storing key ordered list of objects. Page 13, lines 1-19 of Platt are referenced. Furthermore, the second argument, pertaining to sending a query to only one of the servers based upon the data content of the query as in element (b) of applicant's claim 1, was also rejected. The Examiner reasoned that Platt clearly states that a query is sent by the optimizer to only one of the servers in the case where the database is homogenous. Page 15, lines 32-38 of Platt are referenced. Applicants may or may not agree with the conclusions drawn by the Examiner, but have amended independent claims 1 and 16 to overcome the rejection of the Office Action mailed on August 2, 2002. Claims 1 and 16, as now amended, and claims 2-15 and 17-45, as previously pending, are clearly and patentably distinguishable over Platt as reasons are cited below.

In the cited reference to Platt, a system is taught that requires and uses metadata. For example, Figures 4A, and 4C all show rotating storage devices 430 used to hold metadata. Figures 5A and 5B both show rotating storage devices used to hold metadata. Furthermore, when discussing the preferred embodiment starting on page 8, Platt repeatedly refers to the need and use and processing of metadata. Specifically, at page 8, lines 4-7. Platt teaches "This DBMS is used to manage *a storage device 430 having a data dictionary* stored therein. The data dictionary contains *'metadata'* indicating how the data in the database 320 is distributed among the various servers 330, 450, 340 and 350."

In other references cited from Platt, additional teachings of the use of meta data are present. For example, the Parallel Query Decomposer (PQD) middleware refers to the data dictionary (page 8, lines 14-15), interrogates the data dictionary (page 8, lines 31-32 and page 11, lines 25-30), decomposes the original query based on the metadata (page 9, lines 5-15), receives metadata from the data dictionary (page 9, lines 28-29), changes the data dictionary (page 12, lines 1-5), accesses the data dictionary (page 12, lines 10-12), creates the metadata for each partition (page 12, lines 13-15).

Therefore, a data dictionary and metadata are explicit and necessary elements of the system taught by Platt. All of the interrogation, decomposition, referring, transforming, creating, receiving, changing and accessing of data dictionary metadata adds processing steps, requires resources and requires processing.

Claim 1, as now amended, recites a computing system with a key-ordered list of data objects distributed over a plurality of servers which allows discrete parallel processing on said servers, comprising: (a) a *self contained key ordered list of data objects*; (b) a plurality of memories in a plurality of servers, each memory containing a segment of the self contained key-ordered list of the self contained data objects where each segment consists of a contiguous subset of said objects having keys with a specified range; and (c) a query processor which receives queries and, based on data content of the query, directs each received

query to one of said plurality of servers by comparing the data content of the query to the specified range of keys for each segment.

Self contained data objects, by definition, do not have metadata associated with them as is the case with the teachings of Platt. Platt receives a query, performs a search of the metadata and, based on the reply of the metadata search, sends the query to the appropriate server. As is clearly evident, two levels of operations are required and the actual data stored on each server is not self contained because it necessarily requires metadata stored in a different location in order to be accessed.

In direct contrast, claim 1 recites a *self contained* key-ordered list of data objects. In all respects, the data objects on each server are entirely self contained; that is, the data objects do not require references, *i.e.,* metadata, to other data tables, such as data dictionaries, that are shared between segments. Therefore, claim 1, as well as dependent claims 2-15 are allowable for reasons in addition to reasons cited above.

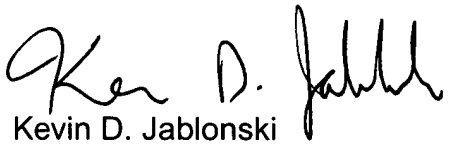
Claim 16 has been amended in a similar way and is allowable, as well as dependent claims 17-45 for reasons in addition to reasons why claim 1 is allowable.

CONCLUSION

In view of the foregoing remarks, applicants submit that all of the claims of the present application as modified are patentably distinguished over the teachings of the cited and applied reference. Thus applicants submit that this application is in condition for allowance. Reconsideration and re-examination of the application and allowance of the claims and passing of the application to issue at an early date are solicited. If the Examiner has any remaining questions concerning this application, the Examiner is invited to contact the applicants' undersigned agent at the number below.

Respectfully submitted,

GRAYBEAL JACKSON HALEY LLP



Kevin D. Jablonski
Registration No.: 50,401
155 108th Ave. NE, Suite 350
Bellevue, WA 98004-5973
Telephone: (425) 455-5575
Facsimile: (425) 455-1046

VERSION TO SHOW MARKED-UP CHANGES TO THE CLAIMS

1. (Amended) A computing system with a key-ordered list of data objects distributed over a plurality of servers which allows discrete parallel processing on said servers, comprising:

(a) a self contained key-ordered list of data objects;

(b) a plurality of memories in a plurality of servers, each memory containing a segment of [a] the self contained key-ordered list of data objects where each segment consists of a contiguous subset of said objects having keys with a specified range; and

(c) a query processor which receives queries and, based on data content of the query, directs each received query to one of said plurality of servers by comparing the data content of the query to the specified range of keys for each segment.

16. (Amended) A method of operating a computing system with a key-ordered list of objects distributed over a plurality of servers which allows discrete parallel processing on said servers, comprising:

(a) creating a self contained key ordered list of data objects;

(b) operating a plurality of servers with a plurality of memories, each memory containing a segment of [a] the self contained key-ordered list of data objects where each segment consists of a contiguous subset of said objects having keys with a specified range; and

(c) operating a query processor which receives queries and, based on data content of the query, directs each received query to one of said plurality of servers by comparing the data content of the query to the specified range of keys for each segment.